Restart strategies for GRASP with path-relinking heuristics

Mauricio G.C. Resende[1] and Celso C. Ribeiro[2]

[1] AT&T Labs Research, Florham Park, NJ 07932-0971, USA,
`mgcr@research.att.com`,
WWW home page: `http://www.research.att.com/~mgcr`
[2] Universidade Federal Fluminense, Niterói, RJ 24210-240 Brazil,
`celso@ic.uff.br`,
WWW home page: `http://www.ic.uff.br/~celso`

Abstract. GRASP with path-relinking is a hybrid metaheuristic, or stochastic local search (Monte Carlo) method, for combinatorial optimization. A restart strategy in GRASP with path-relinking heuristics is a set of iterations $\{i_1, i_2, \ldots\}$ on which the heuristic is restarted from scratch using a new seed for the random number generator. Restart strategies have been shown to speed up stochastic local search algorithms. In this paper, we propose a new restart strategy for GRASP with path-relinking heuristics. We illustrate the speedup obtained with our restart strategy on GRASP with path-relinking heuristics for the maximum cut problem, the maximum weighted satisfiability problem, and the private virtual circuit routing problem.

Keywords: GRASP, path-relinking, restart strategy, experimental algorithmics

1 Introduction

A combinatorial optimization problem is defined by a finite ground set $E = \{1, \ldots, n\}$, a set of feasible solutions $F \subseteq 2^E$, and an objective function $f : 2^E \to \mathbb{R}$. In its minimization version, a global optimum $x^* \in F$ is sought such that $f(x^*) \leq f(x)$, $\forall x \in F$, with each solution being represented by its characteristic vector $x \in \{0, 1\}^{|E|}$. The ground set E, the cost function f, and the set of feasible solutions F are defined for each specific problem.

Metaheuristics are high level procedures for combinatorial optimization that coordinate simple heuristics, such as local search, to find solutions that are of better quality than those found by the simple heuristics alone. Many metaheuristics have been introduced in the last thirty years [10, 12]. Among these, we find genetic algorithms, tabu search, variable neighborhood search, scatter search, iterated local search, path-relinking, and GRASP.

GRASP, or greedy randomized adaptive search procedure, was first introduced in 1989 by Feo and Resende [3]. Path-relinking [11, 13, 25, 26] is an intensification scheme which explores paths in the solution space that connect high-quality solutions. Often, even better-quality solutions can be found in these paths. The hybridization of GRASP with path-relinking adds memory mechanisms to GRASP. It was first proposed by Laguna and Martí [15] and has become the standard way to implement effective GRASP heuristics [24, 26].

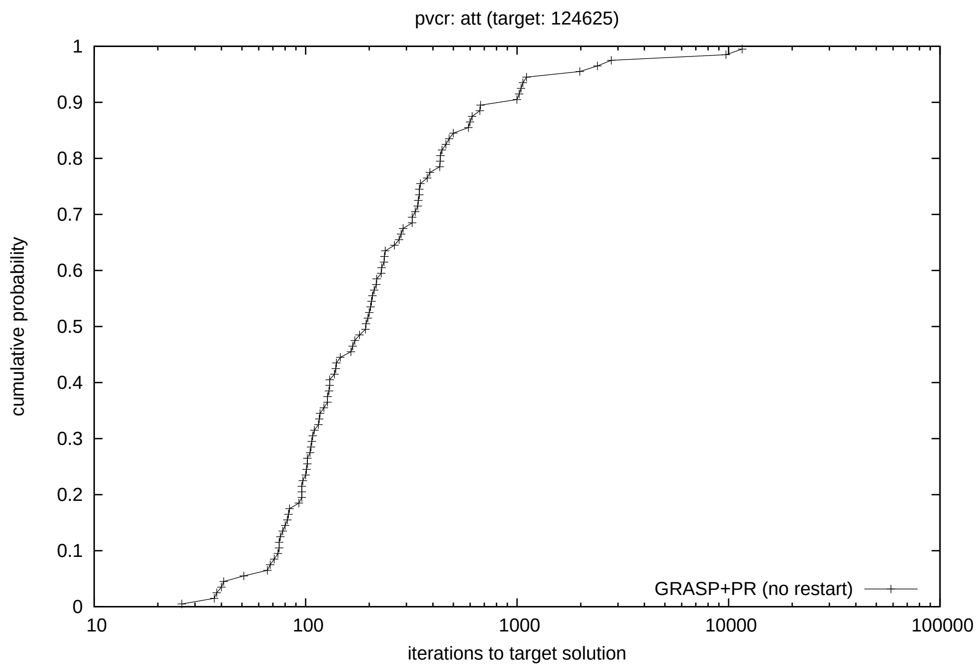

Fig. 1. Typical iteration count distribution of GRASP with path-relinking.

Runtime distributions, or time-to-target plots, display on the ordinate axis the probability that an algorithm will find a solution at least as good as a given target value within a given running time, shown on the abscissa axis. They are constructed by independently running the algorithm a number of times, each time stopping when the algorithm finds a solutions at least as good as a given target solution. Runtime distributions have been advocated as a way to characterize the running times of stochastic algorithms for combinatorial optimization. Runtime distributions are, however, machine dependent. A machine independent alternative is the iteration count distribution. Similar to runtime distributions, they show the probability that an algorithm will find a solution at least as good as a given target value within a given number of iterations. We should note that while an iteration count distribution characterizes the behavior of a given combinatorial optimization algorithm, these distributions should not be used to compare two algorithms that have different running times per iteration, as e.g.

to compare GRASP with GRASP with path-relinking. For those comparisons runtime distributions are more appropriate.

Figure 1 shows a typical iteration count distribution for a GRASP with path-relinking heuristic. The reader will note in this example that for most of the independent runs, the algorithm finds the target solution in relatively few iterations: 25% of the runs take at most 101 iterations; 50% take at most 192 iterations; and 75% take at most 345. However, some runs take much longer: 10% take over 1000 iterations; 5% over 2000; and 2% over 9715 iterations. The longest run took 11607 iterations to find a solution as good as the target. These long tails contribute to a large average iteration count as well as to a high standard deviation. The objective of this paper is to propose strategies to reduce the tail of the distribution, consequently reducing the average iteration count and its standard deviation.

Consider again the distribution in Figure 1. With 25% probability the run will take over 345 iteration. By restarting the algorithm after 345 iterations, the new run will finish by iteration 690 with 75% probability. The probability that the algorithm will still be running after k periods of 345 iterations is $1/(4^k)$. In the example of Figure 1, the probability that the algorithm will be running after 1725 iterations will be about 0.1%, i.e. much less than the 5% probability that the algorithm will take over 2000 iterations without restart.

Restart strategies for speeding up stochastic local search algorithms were first proposed by Luby et al. [16]. They define a restart strategy as an infinite sequence of time intervals $S = \{\tau_1, \tau_2, \tau_3 \ldots\}$ which define epochs $\tau_1, \tau_1+\tau_2, \tau_1+\tau_2+\tau_3, \ldots$ when the algorithm is restarted from scratch, i.e. using a new random number generator seed. Luby et al. [16] prove that the optimal restart strategy uses $\tau_1 = \tau_2 = \cdots = \tau^*$, where τ^* is a constant. Restart strategies in metaheuristics have been addressed in [1, 14, 18, 19, 27]. Some recent work on restart strategies can be found in [28, 29]. To the best of our knowledge no paper to date has addressed restart strategies for GRASP or GRASP with path-relinking heuristics.

The paper is organized as follows. In Section 2, we review basic concepts of GRASP with path-relinking. Simple restart strategies for GRASP with path-relinking are proposed in Section 3. Computational results are summarized in Section 4 and concluding remarks are made in Section 5.

2 GRASP with path-relinking

A GRASP [3, 4] is a multi-start metaheuristic where at each iteration a greedy randomized solution is constructed to be used as a starting solution for local search. If the greedy randomized solution is infeasible, a repair routine may need to be called to make it feasible before local search is applied. The best local minimum found over all GRASP iterations is output as the solution. See [4, 20, 21, 24] for surveys of GRASP and [7–9] for annotated bibliographies.

GRASP iterations are independent, i.e. solutions found in previous GRASP iterations do not influence the algorithm in the current iteration. The use of previously found solutions to influence the procedure in the current iteration

```
begin GRASP+PR
   Elite set E ← ∅;
   while stopping criterion not satisfied do
       x ← RandomizedGreedy(·);
       if x is infeasible then x ← Repair(x);
       x ← LocalSearch(x);
       Select y ∈ E at random;
       z ← PathRelinking(x, y);
       Insert x in E if it meets quality and diversity criteria;
       Insert z in E if it meets quality and diversity criteria;
   end-while;
   return z ← argmin{c(x) | x ∈ E};
end
```

Fig. 2. Pseudo-code of GRASP with path-relinking for minimization

can be thought of as a memory mechanism. One way to incorporate memory into GRASP is with path-relinking [11, 13, 25]. In GRASP with path-relinking (GRASP+PR) [15, 23], an elite set of diverse good-quality solutions is maintained to be used during each GRASP iteration. After a solution is produced with greedy randomized construction and local search, that solution is combined with a randomly selected solution from the elite set using the path-relinking operator. The solution of the local search as well as the best of the combined solutions from path-relinking are candidates for inclusion in the elite set and each is added to the elite set if it meets quality and diversity criteria. Figure 2 shows pseudo-code for a GRASP with path-relinking heuristic.

3 Restart strategy for GRASP with path-relinking

Recall that the optimal restart strategy proposed by Luby et al. [16] uses equal time intervals $\tau_1 = \tau_2 = \cdots = \tau^*$ between restarts. Implementing such a strategy may be difficult in practice because it requires inputting the constant value τ^*. Since we have no a priori information about the runtime distribution of the heuristic for the optimization problem under consideration, we run the risk of choosing a value of τ^* that is either too small or too large. On the one hand, a value that is too small can cause the restart-variant of the heuristic to take much longer to converge than the no-restart variant. On the other hand, a value that is too large may never restart, causing the restart-variant of the heuristic to take as long to converge as the no-restart variant.

A characteristic with less variation between heuristic/instance/target triples than run times is the number of iterations between improvements of the incumbent (or best so far) solution. We propose the following *restart strategy*: Keep track of the last iteration when the incumbent solution was improved and restart the GRASP with path-relinking heuristic if κ iterations have gone by without improvement. We shall call such a strategy restart(κ). This strategy is illustrated in Figure 3 which shows the average time to find a cut of weight at least 554 for max-cut instance *G12* [6] as a function of the restart parameter κ. For each restart parameter, we ran the algorithm 100 times to compute each average. The figure shows that best values of κ are between 200 and 1000 since it is in that range that the average time to target solution is smallest.

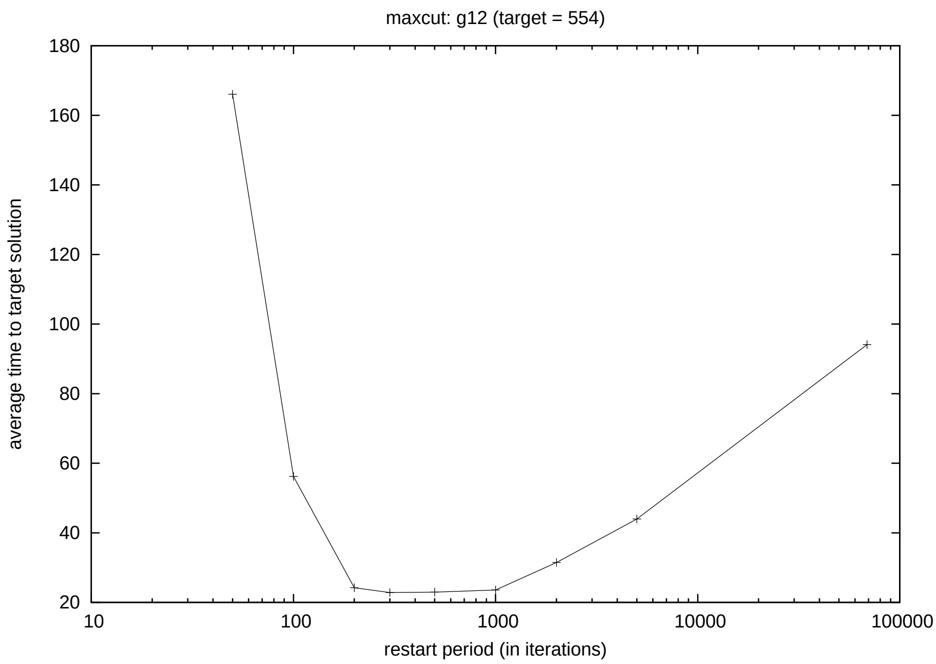

Fig. 3. Average time to find a cut of weight at least 554 for max-cut instance *G12* as a function of the restart parameter κ. The figure shows that the best values are in the range from 200 to 1000 iterations.

Restarting GRASP with path-relinking requires emptying out the elite set, discarding the incumbent, and starting a new iteration with a new seed for the random number generator. In practice, one would also input a maximum number of restarts and store the overall incumbent (over all restarts) to output as the solution. In the experiments in Section 4 we do not do this since we run the heuristics until a solution as good as the target solution is found, i.e. the overall incumbent is the incumbent of the last restart period.

While this strategy also requires us to input a value for parameter κ we will see in the next section that even for heuristic/instance/target triples that differ significantly with respect to runtime distributions, a limited number of values for

κ almost always achieves the desired result, i.e. to reduce the average iteration count as well as its standard deviation.

4 Experiments

In this section we present some preliminary computational results with our restart strategies for GRASP with path-relinking heuristics. We consider three GRASP with path-relinking heuristics: for the maximum cut problem [6], maximum weighted satisfiability [5], and private virtual circuit routing [22]. Each heuristic was implemented using no restart (original GRASP with path-relinking heuristic) and restart strategies: restart(100), restart(500), and restart(1000), which restart, respectively, after 100, 500, and 1000 iterations without improvement in the value of the incumbent.

We consider two instances for each heuristic. For the maximum cut problem we consider instances *G1* and *G12* [6] with target values 11575 and 554, respectively. For the maximum weighted satisfiability problem we consider instances *jnh1* and *jnh304* [5] with target values 420780 and 444125, respectively. Finally, for the private virtual circuit routing problem we consider instances *att* and *fr750* [22] with target values 124625 and 2040000, respectively.

We run each heuristic strategy independently 100 times for each instance, stopping when a solution at least as good as the target is found. For each run the iteration count at termination is recorded. Figures 4, 5, and 6, are iteration count distribution plots for, respectively, the maximum cut, maximum weighted satisfiability, and private virtual circuit routing problems. Table 1 summarizes the experiments. For each instance, the table shows statistics for each of the four strategies (no restart and restart(100), restart(500), and restart(1000)). The statistics are the maximum iteration counts for each quartile of the distribution (maximum number of iterations taken by the fastest 25%, 50%, 75%, and 100% of the runs) as well as the average iteration count and its standard deviation computed over all 100 runs.

We make the following observations regarding the experiment.

- The effect of the restart strategies can be mainly observed in the column corresponding to the fourth quartile of Table 1. The entries in this quartile correspond to those in the heavy tails of the distributions. The restart strategies in general did not affect the other quartiles of the distributions, which is a desirable characteristic.
- For all instances, compared to the no-restart strategy, at least one restart strategy was able to reduce the maximum number of iterations, average number of iterations, and the standard deviation of number of iterations.
- In only three strategy / instance pairs was the restart strategy not able to reduce the maximum number of iterations taken by the no-restart strategy. These were restart(1000)/ *jnh1* and restart(1000)/ *jnh403*, and restart(1000)/ *fr750*. In the first two pairs, however, both average number of iterations and standard deviation were reduced. In the case of restart(1000)/ *fr750*, no

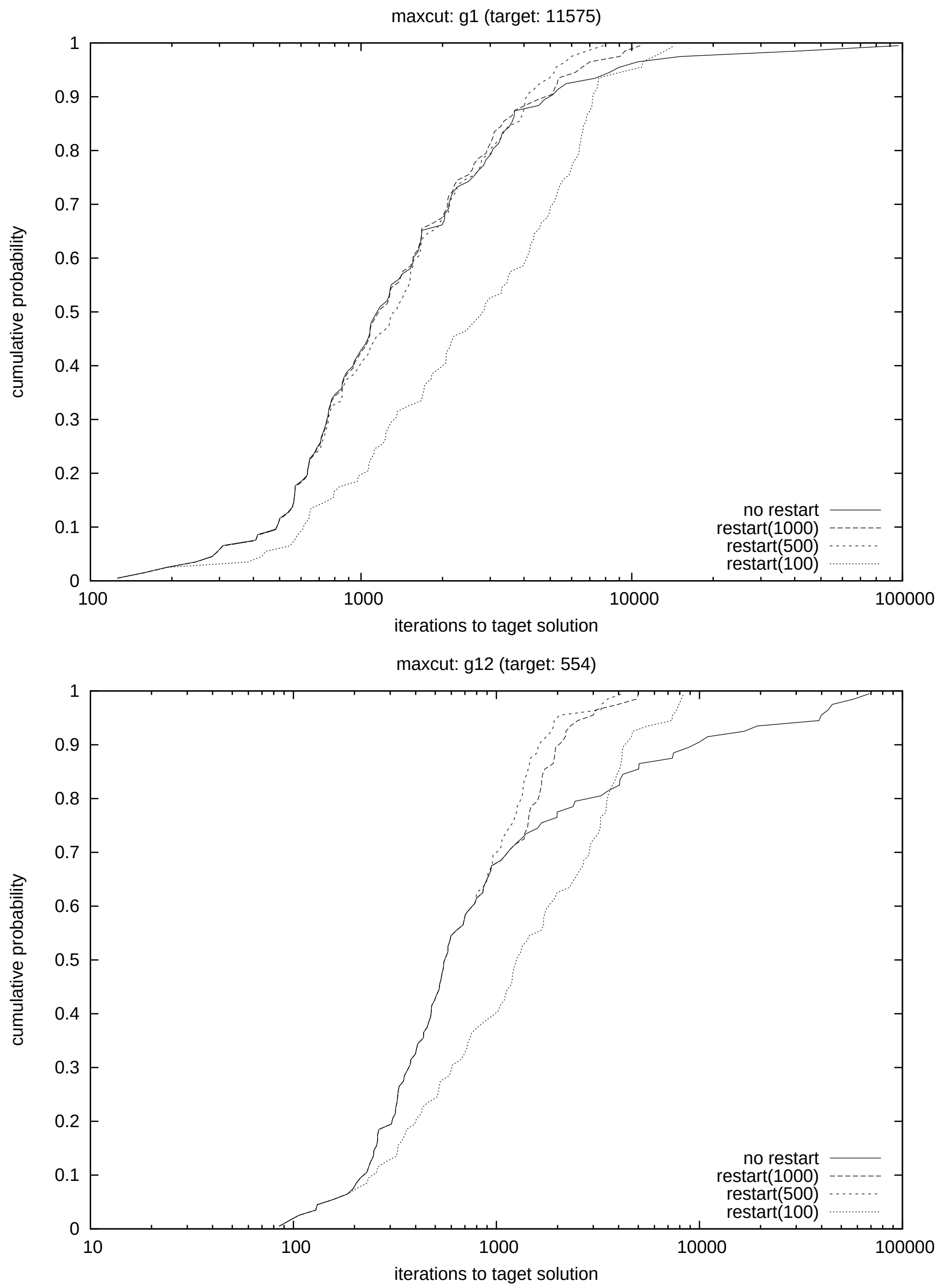

Fig. 4. Runtime distributions comparing the no restart strategy with several restart strategies on maximum cut instances *G1* and *G12*.

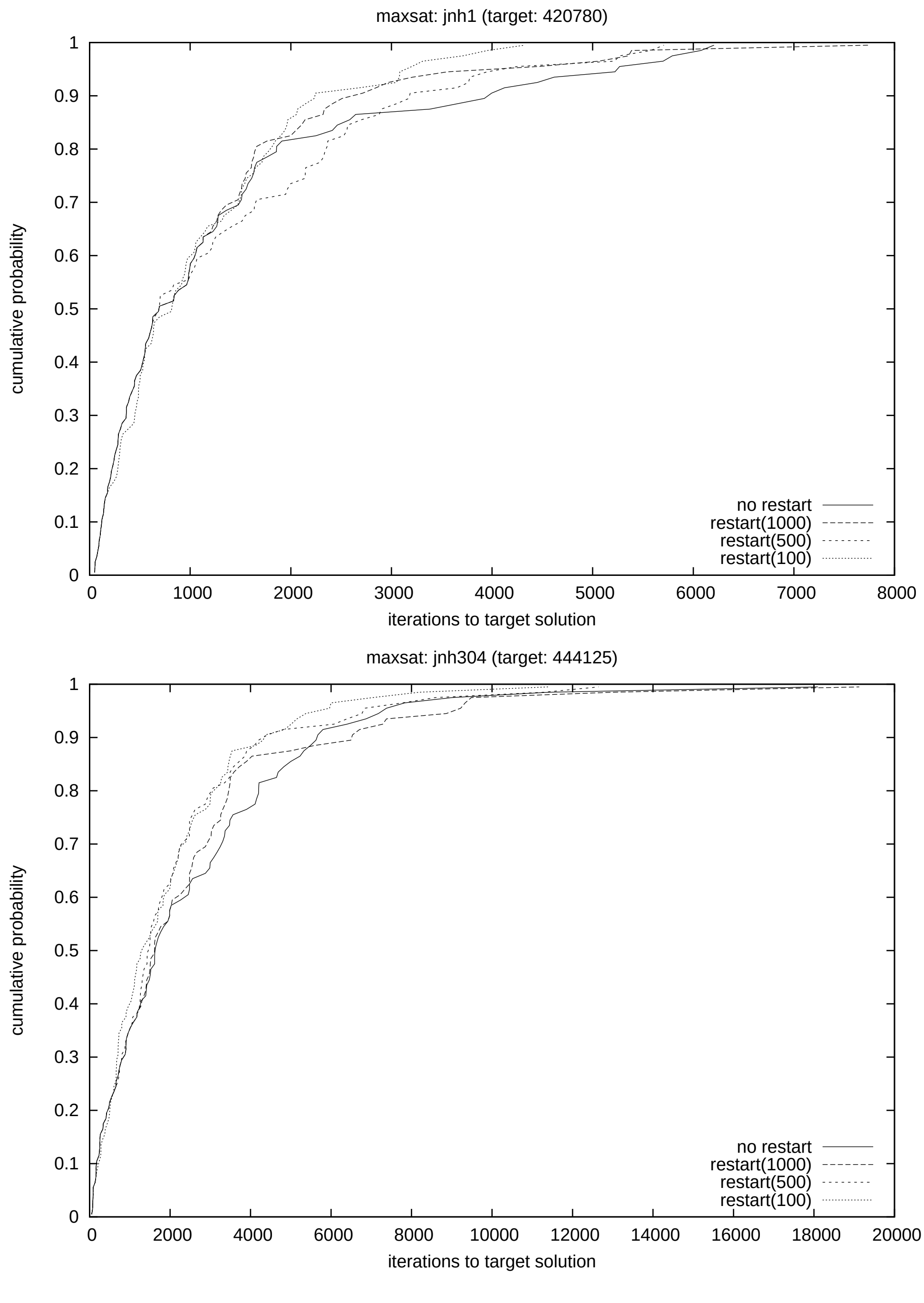

Fig. 5. Runtime distributions comparing the no restart strategy with several restart strategies on maximum weighted satisfiability instances *jnh1* and *jnh304*.

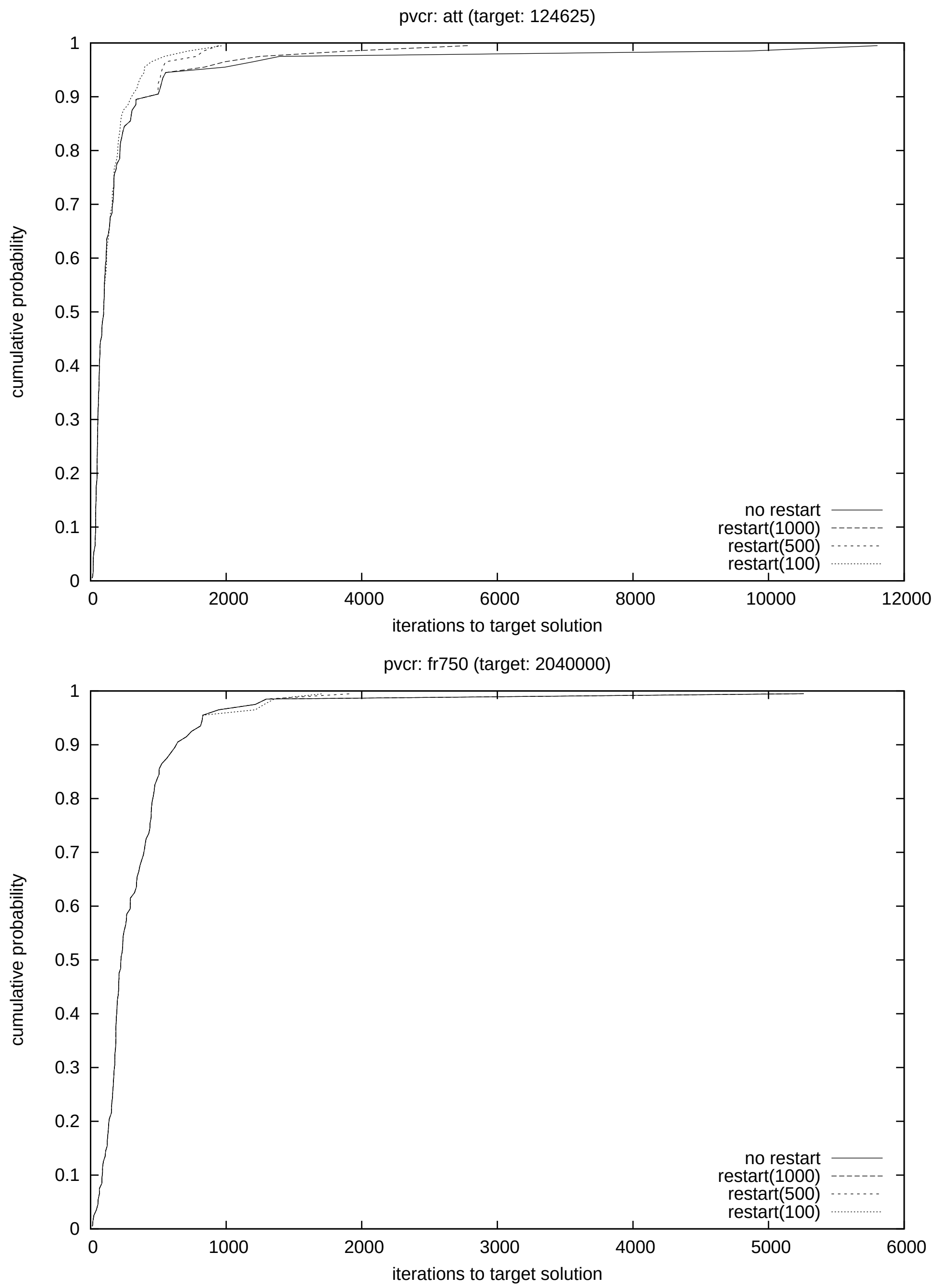

Fig. 6. Runtime distributions comparing the no restart strategy with several restart strategies on private virtual circuit routing instances *att* and *fr750*.

Table 1. Summary of computational results. For each instance and strategy, 100 independent runs were executed, each stopped when a solution as good as a given target solution was found. For each instance / strategy pair, the table shows the distribution of number of iterations by quartile. For each quartile the table shows the maximum number of iterations taken by all runs in that quartile, i.e. the slowest of the fastest 25% (1st), 50% (2nd), 75% (3rd), and 100% (4th) of the runs. The average number of iterations over the 100 runs as well as the standard deviation is given for each instance / strategy pair.

		max itr in quartile					
instance	strategy	1st	2nd	3rd	4th	avg	sdev
maxcut: G1	no restart	708	1145	2610	96763	3331.7	10448.6
	restart(1000)	687	1145	2270	10753	1943.7	2021.9
	restart(500)	708	1292	2404	7849	1850.0	1591.7
	restart(100)	1120	2775	5557	14343	3672.5	3053.9
maxcut: G12	no restart	326	550	1596	68813	4525.1	11927.0
	restart(1000)	326	550	1423	5014	953.2	942.1
	restart(500)	326	550	1152	4178	835.0	746.1
	restart(100)	509	1243	3247	8382	2055.0	2005.9
maxsat: jnh1	no restart	281	684	1611	6206	1319.7	1522.0
	restart(1000)	281	684	1547	7737	1170.6	1317.7
	restart(500)	281	684	2142	5708	1309.2	1363.9
	restart(100)	308	812	1562	4323	1071.3	960.5
maxsat: jnh304	no restart	657	1621	3488	18095	2546.1	2738.2
	restart(1000)	657	1610	3255	19124	2508.0	2957.2
	restart(500)	657	1432	2483	12651	2091.6	2247.3
	restart(100)	605	1266	2558	11390	1929.7	1907.5
pvcr: att	no restart	101	192	345	11607	527.4	1518.2
	restart(1000)	101	192	345	5567	397.9	737.7
	restart(500)	101	192	345	1891	313.6	355.5
	restart(100)	101	192	345	1948	277.9	291.7
pvcr: fr750	no restart	186	223	438	5260	359.0	547.4
	restart(1000)	163	223	438	5260	359.0	547.4
	restart(500)	163	223	438	1924	325.6	287.8
	restart(100)	163	223	438	1717	327.5	288.2

restarts were done since the no-restart strategy never took more than 1000 iterations without improvement of the incumbent.

- In only one strategy / instance pair (restart(100)/ *G1*) was the average number of iterations larger than that of the no-restart strategy. The increase was about 10%.
- In only one strategy / instance pair (restart(1000)/*jnh304*) was the standard deviation of the number of iterations larger than that of the no-restart strategy. The increase was about 8%.
- Compared to the no-restart strategy, restart strategy restart(1000) was able to reduce the maximum number of iterations as well as the average and standard deviation for instances *G1*, *G12*, and *att*. For *jnh1* and *jnh304* it increased the maximum number of iterations. In addition, for *jnh304* it increased the standard deviation. On *fr750* it was not activated a single time.
- Compared to the no-restart strategy, strategy restart(500) was able to reduce the maximum number of iterations as well as the average and standard deviation for all instances. Strategy restart(100) did so, too, for all but one instance (*G1*) where it had a larger average number of iterations than the no-restart strategy.
- Restart strategy restart(500) was clearly the best strategy for instances *G1* and *G12* while restart(100) was the best for instances *jnh1* and *jnh304*. On both private virtual circuit routing instances restart strategies restart(100) and restart(500) were better than restart(1000). Strategy restart(500) reduced the maximum number of generation more than restart(100), while restart(100) reduced the average number of iterations and standard deviation more than restart(500).

5 Concluding remarks

In this paper, we propose new restart strategies for GRASP with path-relinking heuristics. Unlike the strategies considered in the literature, our strategy is based on the number of iterations without improvement of the incumbent solution, This number is monitored and once it reaches a trigger value, the heuristic is restarted by emptying the elite set and incumbent and using a new seed for the random number generator.

We proposed three restart strategies using three different restart trigger values: 100, 500, and 1000. We tested the strategies with GRASP with path-relinking heuristics for maximum cut, maximum weighted satisfiability, and private virtual circuit routing on instances where the average number of iterations of the no-restart variant varied from 359 to 4525 and the maximum number of iterations from 5260 to 96763.

While no restart strategy increased all three performance measures (maximum, average, standard deviation of number of iterations) for a single instance, restart(500) decreased all three measure for all instances while restart(100) increased a single measure for a single instance. Overall, restart(500) was the best strategy.

We must emphasize that these conclusions are valid for these implementations of GRASP with path-relinking on these instances and for these target solution values. Though we conjecture that they are also valid for other implementations, instances, and target values, we will need to carry out further experiments to confirm this. In the full paper, we plan to extend the experiment to a few more GRASP with path-relinking implementations, such as the one for the generalized quadratic assignment problem [17] and for the antibandwidth problem [2] and on a wider range of instances and target solution values.

References

1. D'Apuzzo, M., Migdalas, A., Pardalos, P., Toraldo, G.: Parallel computing in global optimization. In: Kontoghiorghes, E. (ed.) Handbook of parallel computing and statistics. Chapman & Hall / CRC (2006)
2. Duarte, A., Martí, R., Resende, M., Silva, R.: GRASP with path relinking heuristics for the antibandwidth problem. Networks (2010), `http://dx.doi.org/10.1002/net.20418`
3. Feo, T., Resende, M.: A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters 8, 67–71 (1989)
4. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. J. of Global Optimization 6, 109–133 (1995)
5. Festa, P., Pardalos, P., Pitsoulis, L., Resende, M.: GRASP with path-relinking for the weighted MAXSAT problem. ACM J. of Experimental Algorithmics 11, 1–16 (2006), article 2.4
6. Festa, P., Pardalos, P., Resende, M., Ribeiro, C.: Randomized heuristics for the MAX-CUT problem. Optimization Methods and Software 7, 1033–1058 (2002)
7. Festa, P., Resende, M.: GRASP: An annotated bibliography. In: Ribeiro, C., Hansen, P. (eds.) Essays and Surveys on Metaheuristics, pp. 325–367. Kluwer Academic Publishers (2002)
8. Festa, P., Resende, M.: An annotated bibliography of GRASP – Part I: Algorithms. International Transactions on Operational Research 16, 1–24 (2009)
9. Festa, P., Resende, M.: An annotated bibliography of GRASP – Part II: Applications. International Transactions on Operational Research (2009), in press.
10. Gendreau, M., Potvin, J.Y. (eds.): Handbook of Metaheuristics. Springer Science+Business Media, 2nd edn. (2010)
11. Glover, F.: Tabu search and adaptive memory programing – Advances, applications and challenges. In: Barr, R., Helgason, R., Kennington, J. (eds.) Interfaces in Computer Science and Operations Research, pp. 1–75. Kluwer (1996)
12. Glover, F., Kochenberger, G. (eds.): Handbook of Metaheuristics. Kluwer Academic Publishers (2002)
13. Glover, F., Laguna, M., Martí, R.: Fundamentals of scatter search and path relinking. Control and Cybernetics 39, 653–684 (2000)
14. Kautz, H., Horvitz, E., Ruan, Y., Gomes, C., Selman, B.: Dynamic restart policies. In: Proceedings of the National Conference on Artificial Intelligence. pp. 674–681. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2002)
15. Laguna, M., Martí, R.: GRASP and path relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing 11, 44–52 (1999)
16. Luby, M., Sinclair, A., Zuckerman, D.: Optimal speedup of Las Vegas algorithms. Information Processing Letters 47, 173–180 (1993)

17. Mateus, G., Resende, M., Silva, R.: GRASP with path-relinking for the generalized quadratic assignment problem. Journal of Heuristics pp. 1–39 (2010), `http://dx.doi.org/10.1007/s10732-010-9144-0`
18. Nowicki, E., Smutnicki, C.: An advanced tabu search algorithm for the job shop problem. Journal of Scheduling 8(2), 145–159 (2005)
19. Palubeckis, G.: Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. Annals of Operations Research 131(1), 259–282 (2004)
20. Pitsoulis, L., Resende, M.: Greedy randomized adaptive search procedures. In: Pardalos, P., Resende, M. (eds.) Handbook of Applied Optimization, pp. 168–183. Oxford University Press (2002)
21. Resende, M., Ribeiro, C.: Greedy randomized adaptive search procedures. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics, pp. 219–249. Kluwer Academic Publishers (2002)
22. Resende, M., Ribeiro, C.: A GRASP with path-relinking for private virtual circuit routing. Networks 41(1), 104–114 (2003)
23. Resende, M., Ribeiro, C.: GRASP with path-relinking: Recent advances and applications. In: Ibaraki, T., Nonobe, K., Yagiura, M. (eds.) Metaheuristics: Progress as Real Problem Solvers, pp. 29–63. Springer (2005)
24. Resende, M., Ribeiro, C.: Greedy randomized adaptive search procedures: Advances and applications. In: Gendreau, M., Potvin, J.Y. (eds.) Handbook of Metaheuristics, pp. 281–317. Springer Science+Business Media, 2nd edn. (2010)
25. Resende, M., Ribeiro, C., Martí, R., Glover, F.: Scatter search and path-relinking: Fundamentals, advances, and applications. In: Gendreau, M., Potvin, J.Y. (eds.) Handbook of Metaheuristics, pp. 87–107. Springer Science+Business Media, 2nd edn. (2010)
26. Ribeiro, C.C., Resende, M.G.C.: Path-relinking intensification methods for stochastic local search algorithms. Tech. rep., AT&T Labs Research, Florham Park, NJ 07932 (2010), `http://www.research.att.com/~mgcr/docs/spr.pdf`
27. Sergienko, I.V., Shilo, V.P., Roshchin, V.A.: Optimization parallelizing for discrete programming problems. Cybernetics and Systems Analysis 40(2), 184–189 (2004)
28. Shylo, O.V., Middelkoop, T., M.Pardalos, P.: Restart strategies in optimization: Parallel and serial cases. Parallel Computing 37, 60–68 (2011)
29. Shylo, O.V., Prokopyev, O.A., Rajgopal, J.: On algorithm portfolios and restart strategies. Operations Research Letters 39, 49–52 (2011)